下载APP

04 | 指日可待：一步一步搭建秒杀系统（下）

2021-10-01 佘志东

《手把手带你搭建秒杀系统》 课程介绍 >

讲述：佘志东
时长 12:14 大小 11.22M

你好，我是志东，欢迎和我一起从零打造秒杀系统。

在上节课中，我们完成了 3 个项目的搭建，并给项目做了大致的最终目标职能划分。那么接下来我们就可以对秒杀业务的流程做一个详细的梳理了，给出要实现的交互逻辑，然后按照交互需求，提炼出各个项目具体要提供的接口，之后按照各个接口要实现的功能，去具体**开发我们的业务代码**，最终实现秒杀活动开展的全闭环。话不多说，现在就开始吧。

秒杀业务流程梳理

根据我们之前对秒杀业务的介绍，一场完整的秒杀活动的大概流程是这样的，我们一起梳理一下。

1. 运营人员在秒杀系统的运营后台，根据指定商品，创建秒杀活动，指定活动的开始时间、结束时间、活动库存等。

2. 活动开始之前，由秒杀系统运营后台 worker，将活动商品的标识更改为秒杀标识。

3. 用户进入到商详页面时，系统会判断当前商品标识，如果是秒杀标识，则去查询当前商品的秒杀活动数据，判断是否正式开始，即通过商品标识 + 活动时间来判断活动是否真正开始。如果活动时间还没有到，页面可以是禁售展示，也可以是倒计时展示，或者是按正常价格售卖，这个可以按实际业务需求来定。

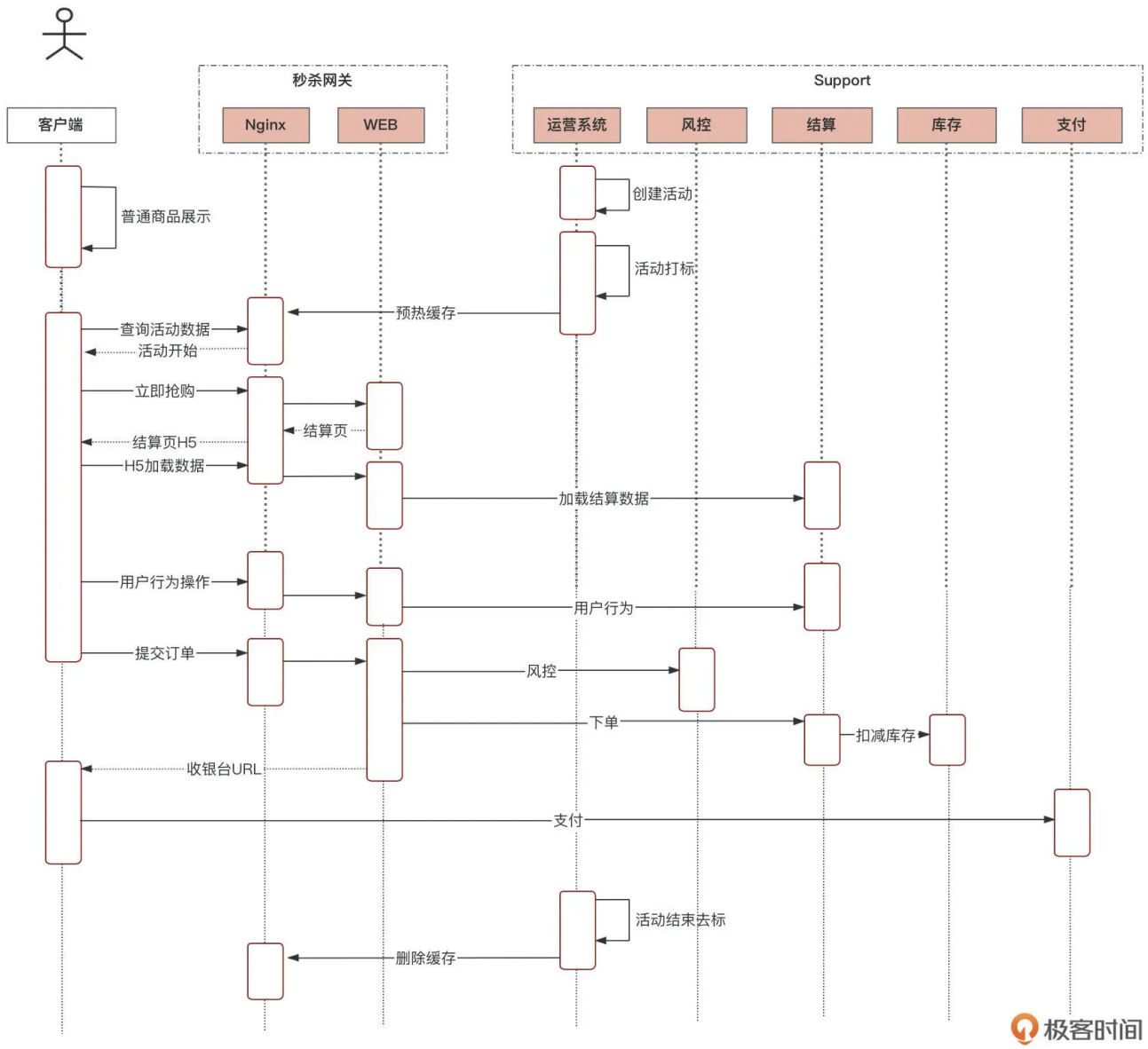
4. 当活动已经开始，用户进入商详页，可以看到立即抢购的按钮，这里我们可以通过增加一些逻辑判断来限制按钮是否可以点击，比如是否设置了抢购用户等级限制，是否还有活动库存，是否设置了预约等等。如果都没限制，用户可以点击抢购按钮，进入到秒杀结算页。

5. 在结算页，用户可更改购买数量，切换地址、支付方式等，这里的结算元素也需要按实际业务来定，更复杂的场景还可以支持积分、优惠券、红包、配送时效等，并且这些都会影响最终价格的计算。

6. 确认无误后，用户提交订单，在这里后端服务可以调用风控、限购等接口，来完善校验，都通过之后，完成库存的扣减和订单的生成。如果结算页支持了第 5 步中提到的一些虚拟资产，则还需要做对应的抵扣。

7. 订单完成后，根据用户选择的支付方式跳转到对应的页面，比如在线支付就跳转到收银台，货到付款的话，就跳到下单成功提示页。

整个时序图如下：



这样一来，秒杀业务从开始到用户抢购，到最后的活动结束关闭，整个流程就形成**闭环**了。当然上面列举的也只是主要的流程，实际业务可以在不同节点依据实际需求添加不同的业务功能，这个你可以灵活调整。

系统提供接口梳理

从上面的时序图中，我们可以非常清楚地归纳出秒杀系统需要提供的**主要接口**：

1. 活动数据查询接口：查询活动相关信息，包括开始、结束时间等。

2. 进结算页页面（H5）接口：结算页 H5，并通过 Ajax 异步加载结算页数据。

3. 结算页页面初始化渲染所需数据的接口：大体包括活动信息、商品信息、结算信息（用户的地址、虚拟资产、价格等等）。

4. 结算页页面用户行为操作接口：支持地址列表查看和选择，虚拟资产的查看和使用等等，并在操作后更新页面价格相关信息。

5. 结算页提交订单接口：支持秒杀活动商品下单。

当然这是秒杀网关系统所需要提供的接口，但是要完整地实现整个秒杀功能，我们还得需要以下功能。而这些个功能点，不需要做到秒杀的主流程系统里，一般都有秒杀的运营系统来提供**相应能力**，简列如下：

1. 秒杀活动的创建：创建秒杀活动，主要要素包括活动名称、参加活动的商品、活动库存、活动单次限购数量、活动开始时间、活动结束时间。

2. 秒杀活动的查看：查看活动信息、活动状态等。

3. 秒杀活动的开始：一般活动都是提前创建，并在活动即将开始之前几分钟，自动更改活动商品标识，这样商详页就能区分出当前商品是普通商品还是秒杀商品了，然后执行不同的业务分支逻辑。

4. 秒杀活动的结束：活动时间到期或者运营人员手动关闭，并将活动商品的秒杀标识去掉。

同时为了我们整个秒杀功能的展示，我们还需要有模拟商品信息的查询（后台查看），客户端商品的详情页面（秒杀入口）以及下单成功后的收银台页面，这 3 个一般情况下不属于秒杀整个系统的功能范畴了，这里仅仅是为了更好地展示秒杀的整个链路所做的模拟。

我们需要将所有网关入口都封闭在 demo-nginx（本地为了模拟一些业务功能，会将部分入口放在 demo-web，方便访问）。而所有操作数据库的入口都放在 demo-support，并且为了模拟整个秒杀的流程业务，我们需要以下 3 张表来支撑：

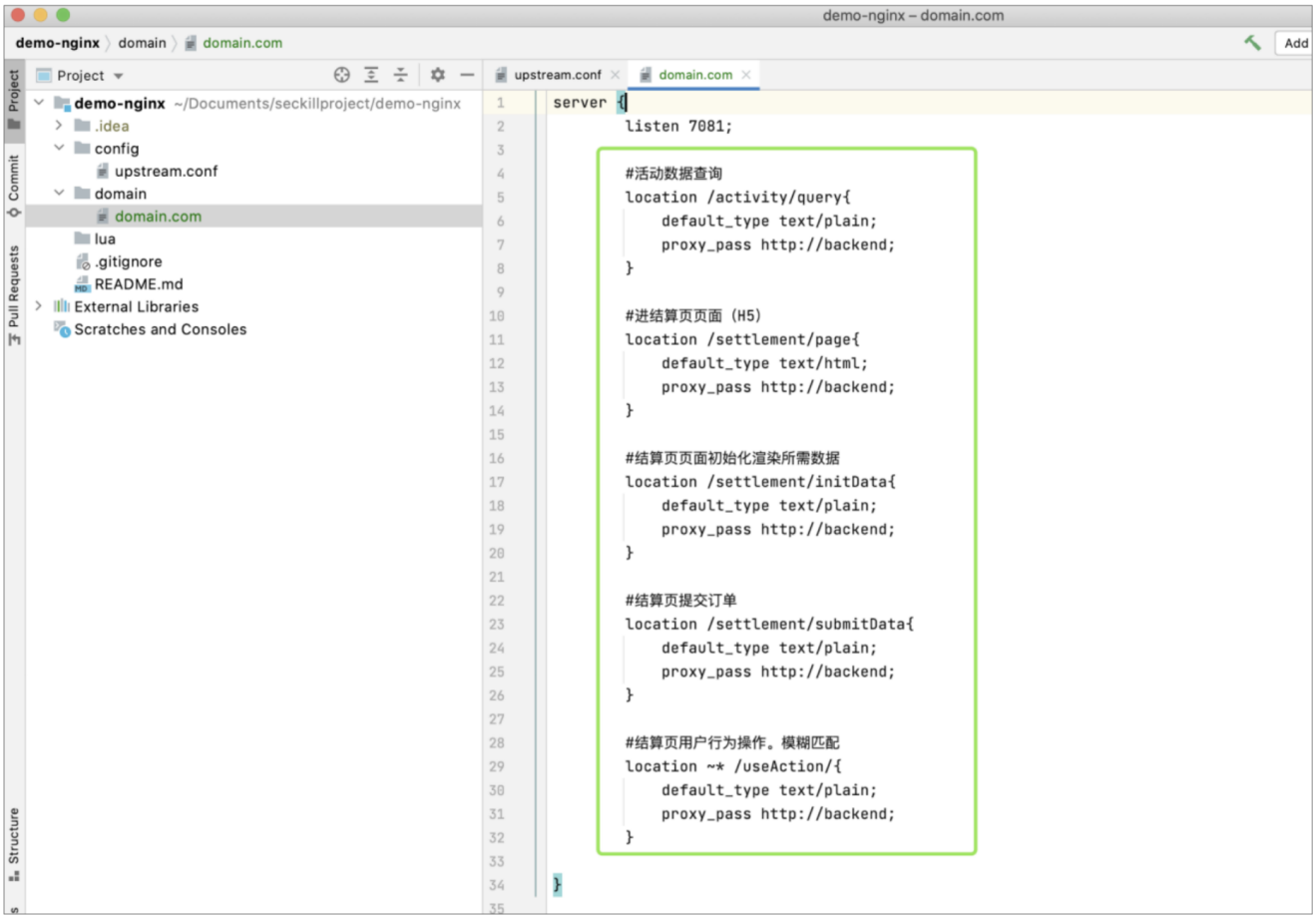
表名	功能描述	所属系统	备注
商品表	存储商品信息	正常不属于秒杀系统，这里做模拟	主要包括商品名称、图片、正常价格，商品标识
秒杀活动表	存储活动信息	一般是秒杀的运营系统来维护	主要包括活动名称、活动商品、活动库存、开始结束时间等，这里一般活动库存和真实库存会区分开，如果有限购，活动库存是可以配置在限购系统中的
订单表	存储下单数据	秒杀依赖系统	主要包括订单号、用户 ID、商品 ID 等

一切就绪，接下来，我们就根据上面的设计，开始投入到实际的开发中去吧。

秒杀业务的实现

这里，详细的开发细节就不说了，因为上面已经理清了所有的接口和对应的能力，实际的代码开发并不复杂，相信对你来说也不是难事。

这里我主要说一下 demo-nginx 的配置，因为我们前面说过，初版先用比较传统的结构来实现秒杀，这样一来主要功能的开发都放在了 demo-web 和 demo-support，所以我们的 demo-nginx 现在还不会有太多的配置更改。只需要将对应的接口 URL 匹配，配置到 domain.com 里即可，实现对应接口请求的接收和分发，修改后文件内容如下：



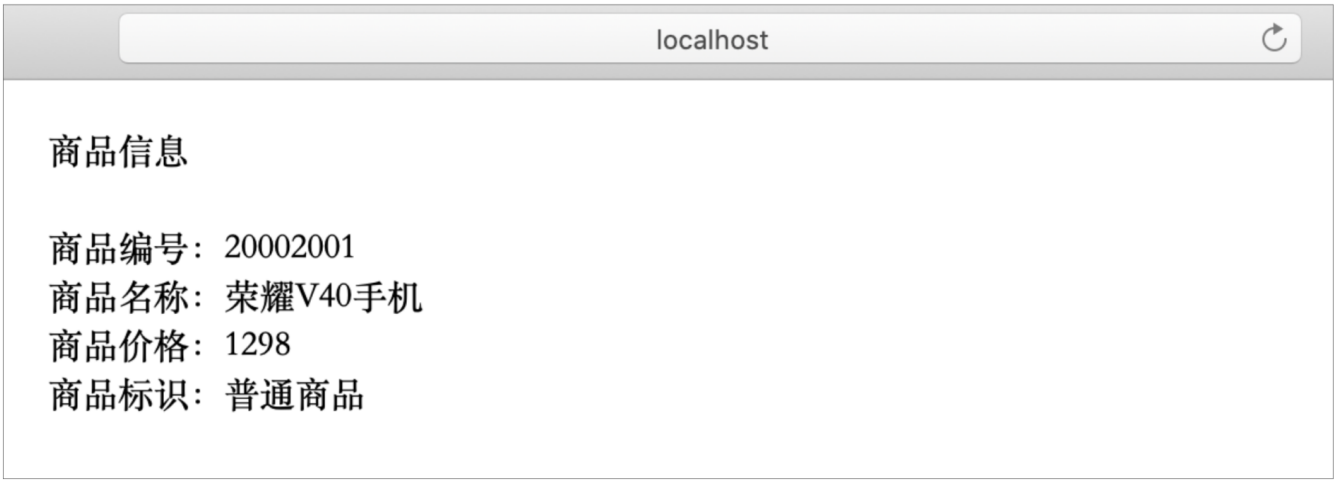
这里结算页用户行为操作的接口，可能是一个或是多个，如果没有特别的处理，可以配置一个模糊匹配，并直接打到后端服务，不需要做额外处理。（其中 ~* 表示启用正则匹配，同时忽略字母大小写，如果遇到请求静态资源不到的情况，也可以参照此方式配置）

省略掉中间的开发过程，我们先来看下最终要实现的效果吧，同时你也可以根据下面的交互，去理解一下我们为什么会要求提供上面的那些接口，**其中各个阶段要调用的接口，我都用橘色特别标识出了。**

第一步

我们先在数据库商品表中初始化一条商品信息，用来模拟参加秒杀活动。然后再通过商品信息查询接口，在浏览器页面查看商品的信息（根据商品编号查询），这是个后台功能，主要是看下商品的基本信息，比如名称、价格、商品标识，特别是商品标识在参加秒杀活动前后的一个变化。

这里为了方便，我把入口放在了 demo-web 服务，我们通过 demo-web 的端口号，访问对应的 URL 即可，如下图所示：



第二步

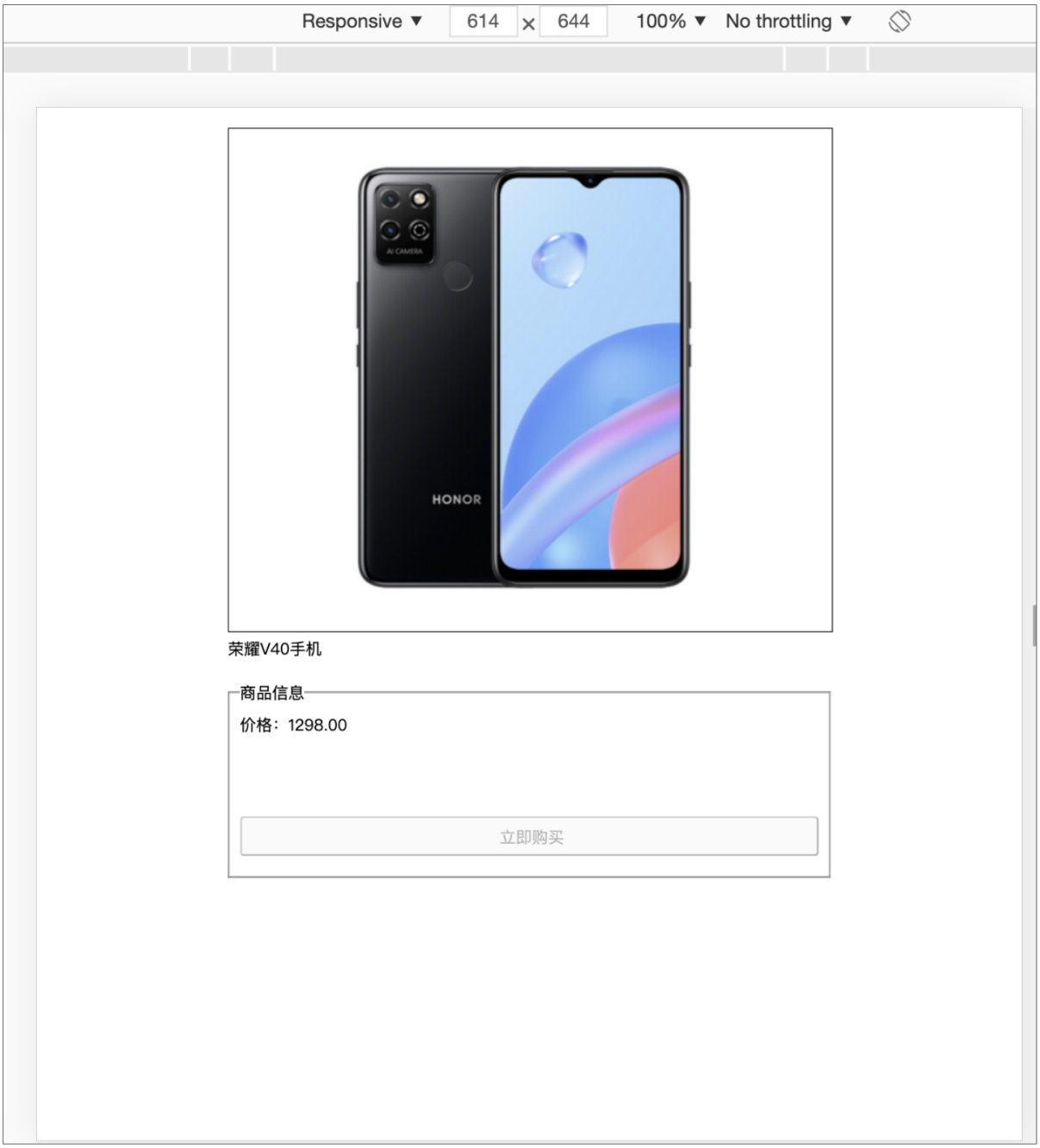
用第一步的商品，创建一个秒杀活动。这个功能正常是放在运营系统里的，通过页面来进行秒杀活动的管理，这里为了方便教学，就直接调用秒杀活动创建接口，来完成秒杀活动的创建，效果如下：

创建完成后，我们调用活动信息查询接口，来查看刚刚创建的活动，页面只是简单地列举了一下活动的主要元素，如下图所示：

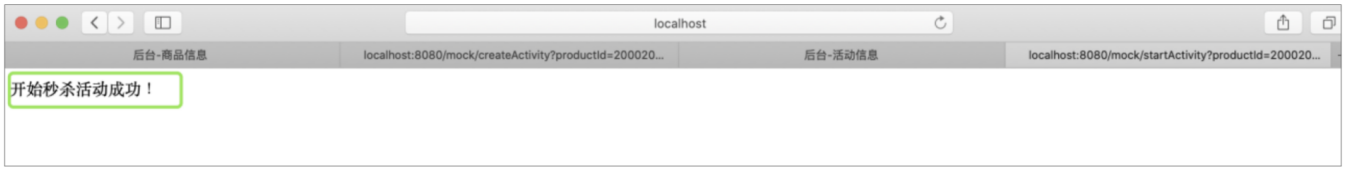
这里我们将参加秒杀活动的商品价格设为了 998，原价是 1298，并且设置了 4 件库存，活动的开始时间是当前时间，结束时间是两天后。正常情况下活动是已经开始了，需要通过 worker 来启动活动，并完成商品标识的修改，但我本地模拟是没有加定时 worker 功能的，想通过 HTTP 接口的方式来触发，所以这里还是显示活动未开始。

第三步

我们先不触发活动开始，我们先模拟客户端进商详页看看现在活动入口页面的样式，同样是通过 URL 方式，直接访问 demo-web，效果如下：



可以看到，活动虽然创建了，但还没有开始，所以这里展示的还是普通商品的名称和价格。下面我们就来手动触发一下，让活动开始，直接调用 demo-web 的活动开始接口：

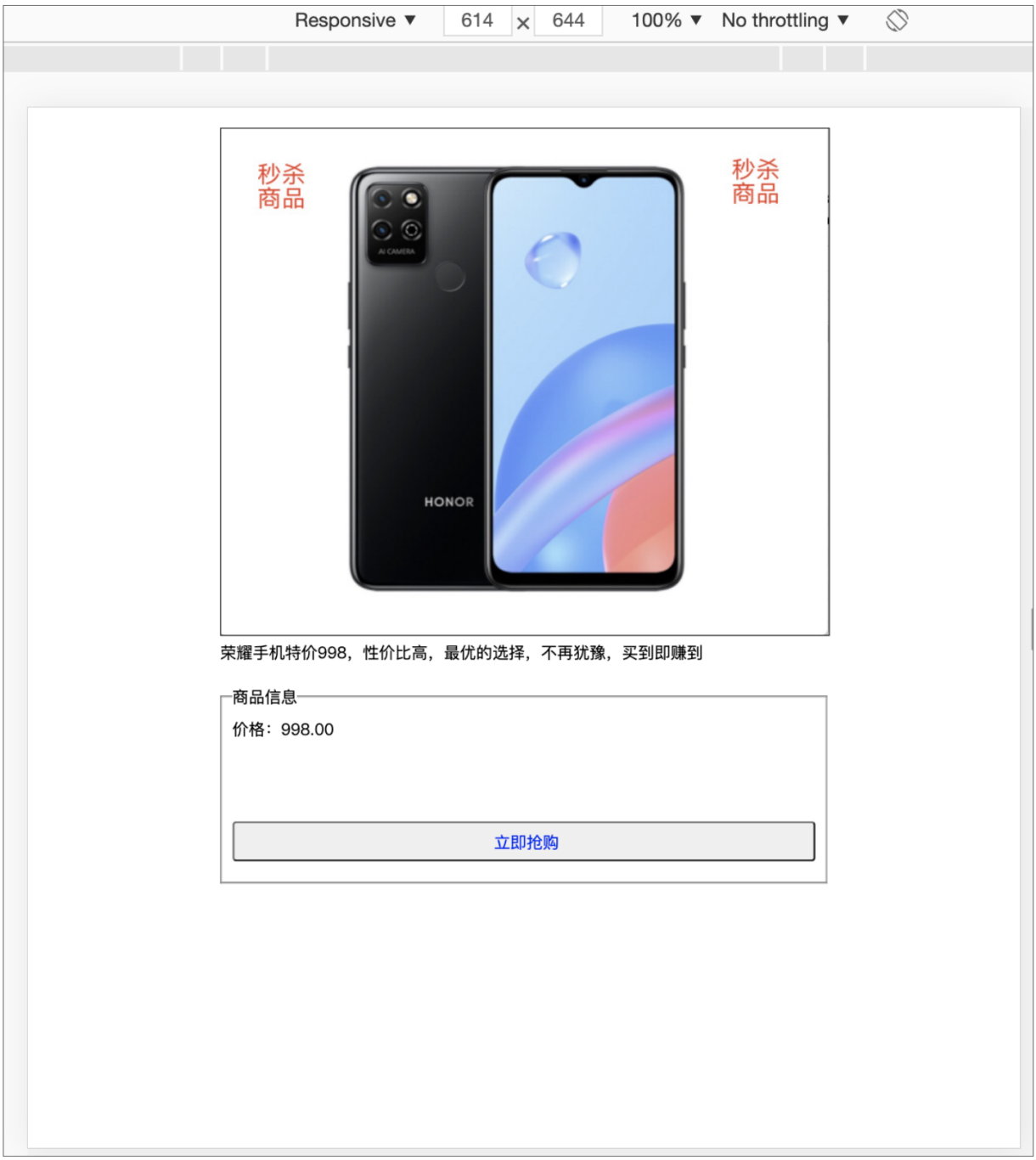


这样活动就触发成功了，这个时候，我们再分别看下活动信息和商品信息，看看都有什么变化：

正如我们所期望的，活动的状态变成了进行中，同时商品标识也变成了秒杀标识。

第四步

秒杀活动开始后，我们再次进入到商详页。



对比之前的商详页，我们可以看到商品的图片、名称、价格展示，都已经变成了活动配置的，同时按钮也变成了立即抢购，并可以点击，说明到这里我们一切都进展的很顺利！

第五步

这时，我们点击立即抢购按钮，去进入到秒杀结算页。这里点击按钮，调用的是进结算页页面（H5）接口，在加载了 HTML 后，JS 通过 Ajax 调用结算页页面初始化渲染所需数据的接口，去渲染页面展示，效果如下：

页面的结算元素，这里只简单地展示了几个，当然你可以根据实际需要去灵活填充。展示中的结算页支持修改购买数量，修改后总金额会随之变化，而且有单次够买数量的限制，同时也允许用户切换支付方式、修改地址等操作，这里做的用户操作，会调用结算页页面用户行为操作接口。

第六步

操作结算元素完成后，我们就可以提交订单了，调用的是结算页提交订单接口。在经过一系列的校验之后，完成库存的预占和订单的生成，并返回收银台的 URL，完成下单成功后的跳转动作，如下图所示：

这里是跳转到了收银台的模拟页面，这时我们再回头看看我们的活动库存：

由 4 件变成了 3 件，说明成功做了库存扣减。后续用户可以继续完成支付相关操作，那么用户的一次抢购行为也就结束了，当然如果用户放弃了支付或是取消了订单，那么需要将预占的库存再恢复回去。

我们这里可以模拟多次抢购，当我们把商品买完时：

再去商详页看一下：

可以发现，展示的虽然还是活动相关数据，但是按钮变成灰色的了，因为没有活动库存了。

第七步

正常情况下，当商品售完时，活动也应该关闭掉了。因为我们没有 worker，所以我们就来手动关闭一下，和触发活动开始一样，调用活动关闭的接口，去关闭活动并将商品的标识恢复成普通商品，如下图所示：

成功后，我们再看下刷新下商详页：

可以看到页面已经恢复到最初的状态了，同时查看下活动信息与商品信息：

活动显示已经结束，商品标识也变回了普通商品，一切都符合预期。

到此，我们就完整地支持了一场秒杀活动的开展，秒杀系统中的一些关键链路也得以完美复现，并顺利地实现了我们设定的那个小目标——实现一个最简的秒杀系统。

小结

在这节课中，我们为了开发一个最简的秒杀系统，针对秒杀流程做了详细的梳理，画出了系统间的交互过程，同时明确列出了整个秒杀系统所需要提供的主要接口和能力，并且为了整个秒杀业务场景的再现，还加入了一些辅助接口，总体归纳如下：

接口名称	所属系统	备注
活动数据查询	demo-nginx,demo-web	入口在 demo-nginx
进结算页页面（H5）	demo-nginx,demo-web	入口在 demo-nginx
结算页页面初始化渲染所需数据	demo-nginx,demo-web	入口在 demo-nginx
结算页页面用户行为操作	demo-nginx,demo-web	入口在 demo-nginx
结算页提交订单	demo-nginx,demo-web	入口在 demo-nginx
秒杀活动的创建	demo-web	正常应放在秒杀的运营系统，入口放在 demo-web 用于模拟
秒杀活动的查看	demo-web	正常应放在秒杀的运营系统，入口放在 demo-web 用于模拟
秒杀活动的开始	demo-web	正常应放在秒杀的运营系统，入口放在 demo-web 用于模拟
秒杀活动的结束	demo-web	正常应放在秒杀的运营系统，入口放在 demo-web 用于模拟
商品信息查询	demo-web	不属于秒杀，入口放在 demo-web 用于模拟
商品详情页面	demo-web	不属于秒杀，入口放在 demo-web 用于模拟
收银台页面	demo-web	不属于秒杀，入口放在 demo-web 用于模拟
基础数据与服务	demo-support	提供数据操作 RPC 接口，包括活动的创建、查询、下单等

有了这些详细设计的指引，我们可以很快速地开发出一个最简的秒杀系统，并能在本地实现“秒杀活动的创建 -> 活动开始的打标 -> 从商详页进秒杀结算页 -> 提交订单 -> 活动的关闭与去标”的完整交互。一方面可以帮助我们熟悉和理解秒杀的整个流程，另一方面也可以让我们以“摸的着”的方式去近距离地接触秒杀系统，并感受秒杀系统的设计之美。

思考题

现在我们已经自己动手实现了整个秒杀业务，也让我们有了去优化秒杀系统的基石。现在你是时候该去认真思考一下，以上提供的每个接口的瓶颈点在哪里，会出现什么问题，以及如何有效应对了。

以上就是这节课的全部内容，欢迎你在评论区和我讨论问题，交流经验！

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 1 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 03 | 指日可待：一步一步搭建秒杀系统（上）

下一篇 05 | 勇于担当：秒杀的隔离策略

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。